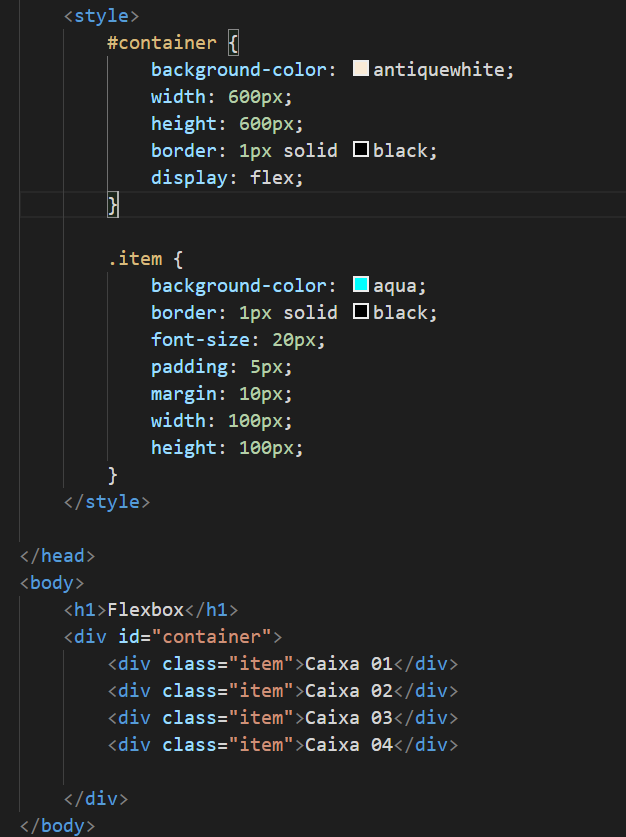
**Flexbox**

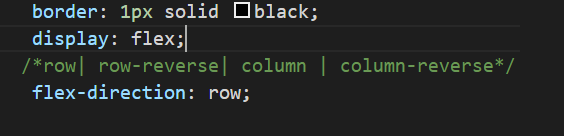
**Vamos montar essa estrutura acrescentando a propriedade display: flex dentro do container, Ex:**

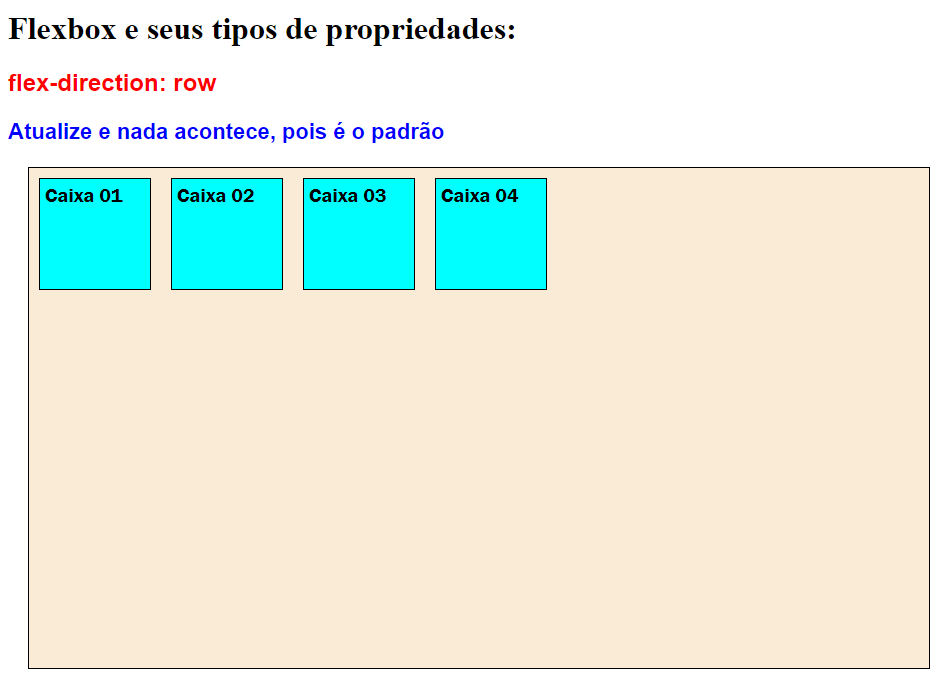
****

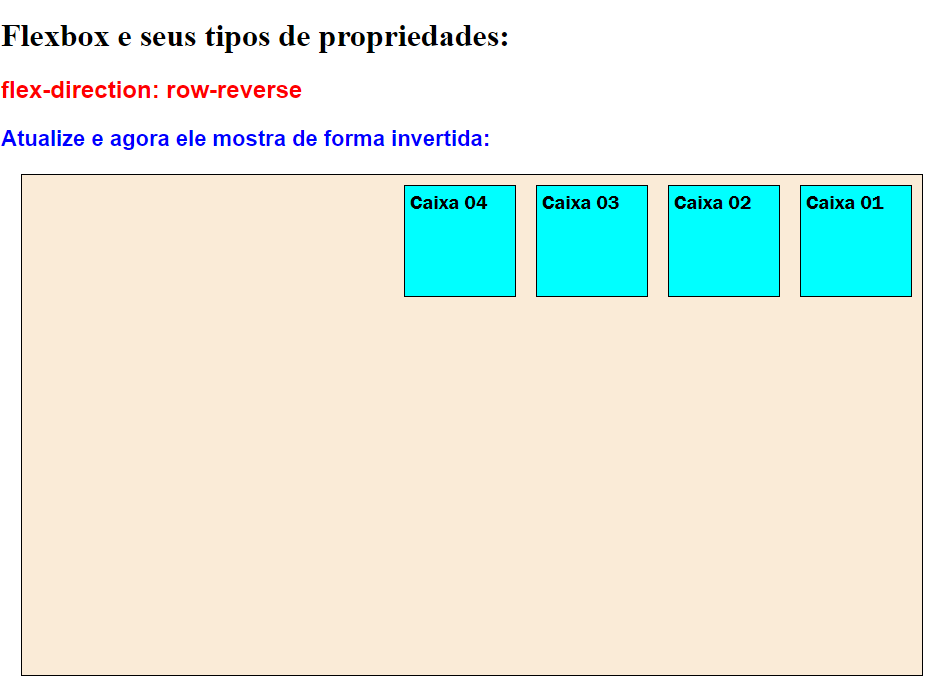
**flex-direction**

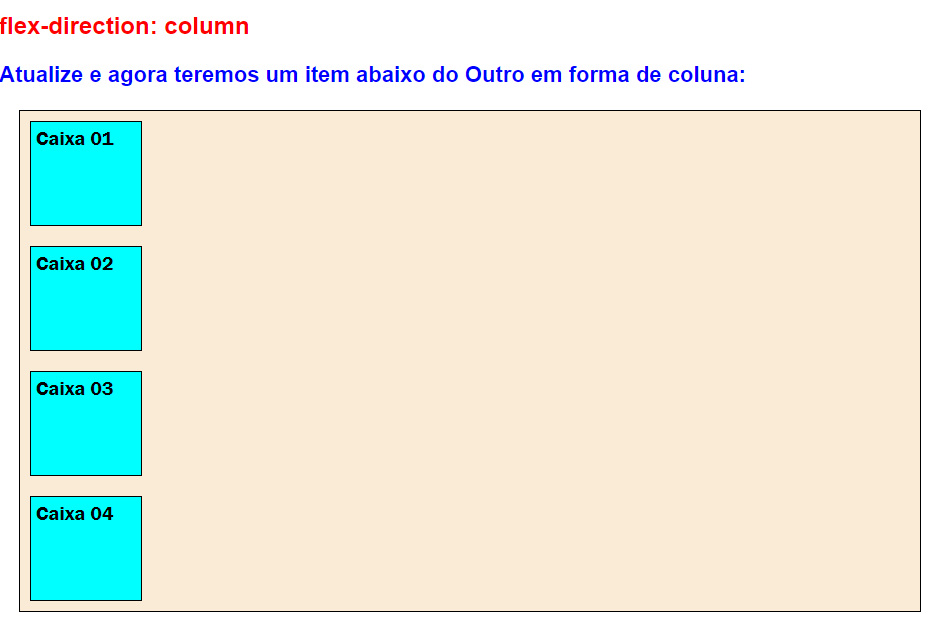
**Os valores dessa propriedade são:**

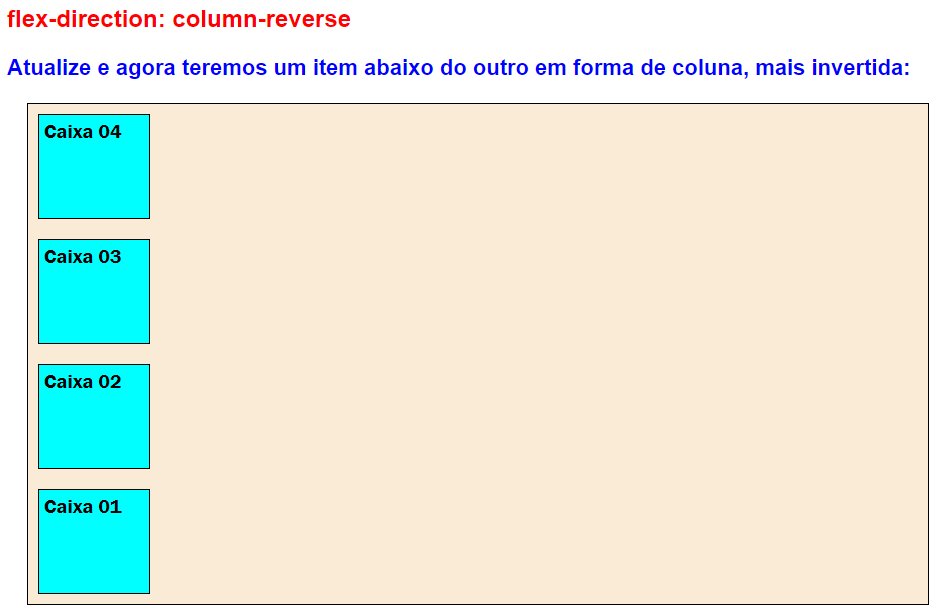
****

****

****

****

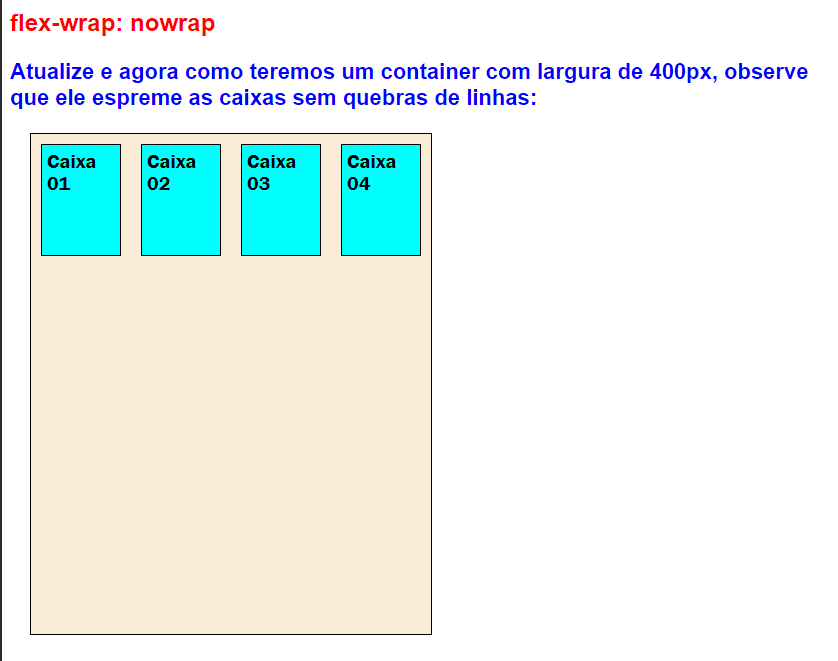
****

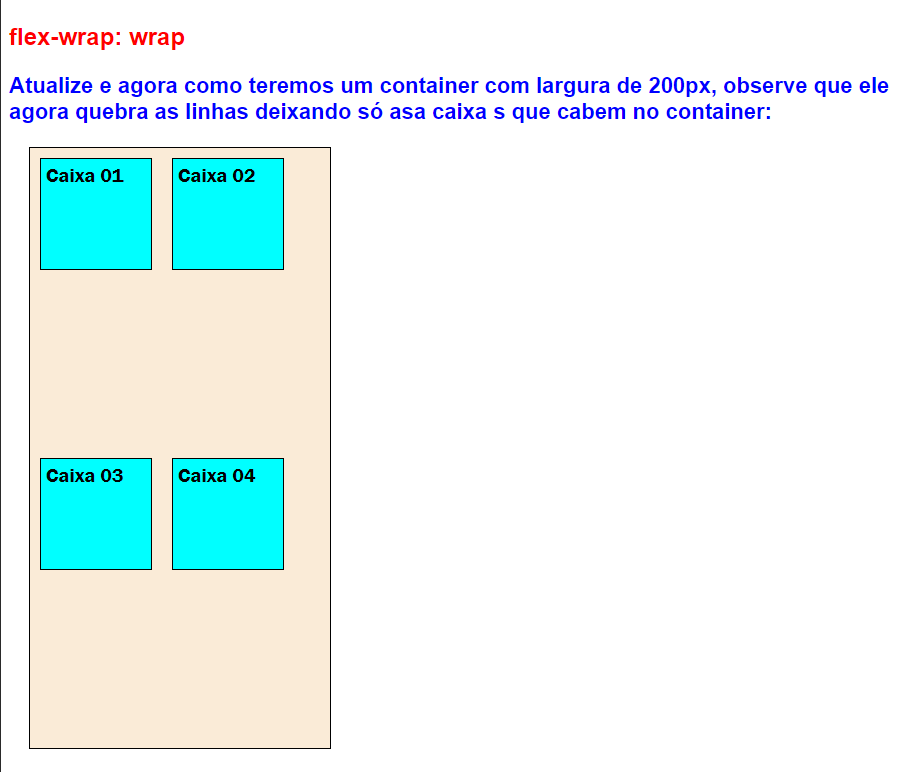
****

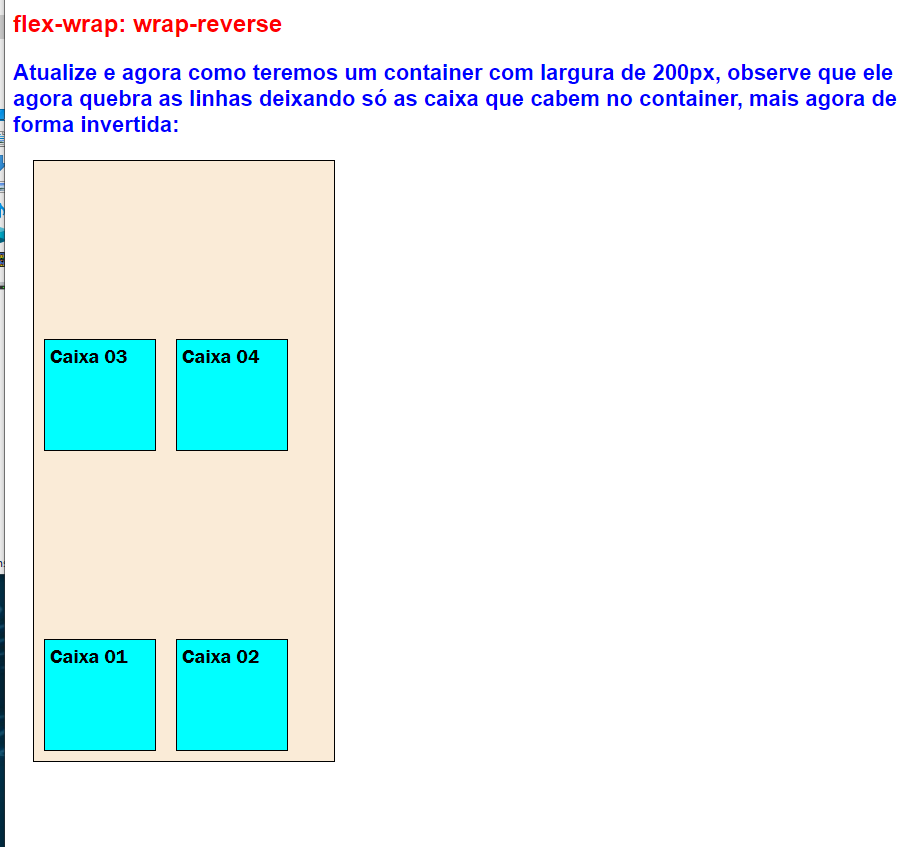
**flex-wrap**

**Os valores dessa propriedade são:**

/\*now-wrap| wrap| wrap-reverse|\*/

****

****

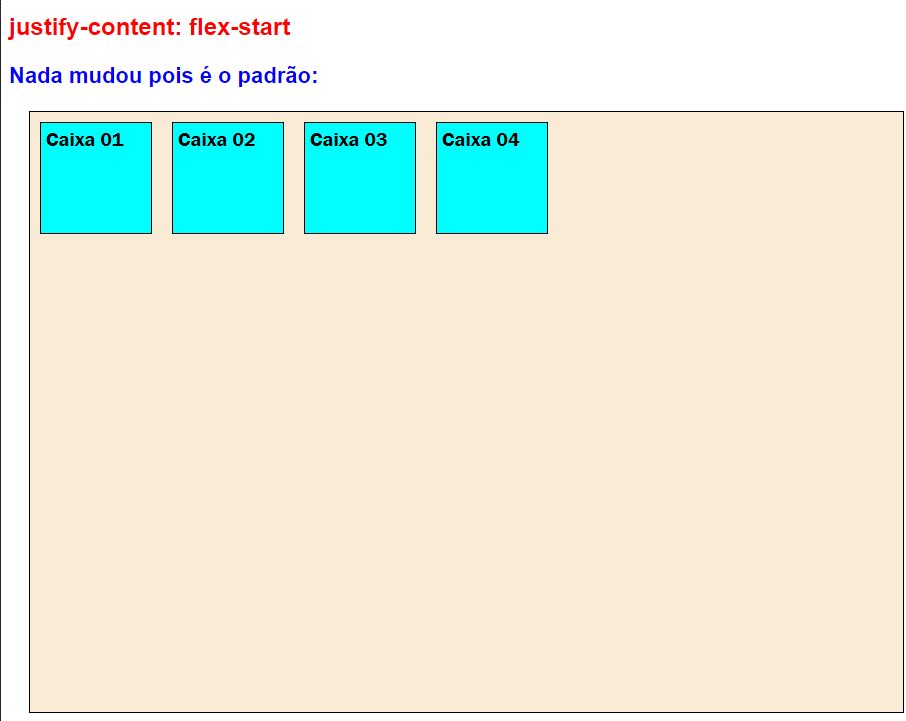
****

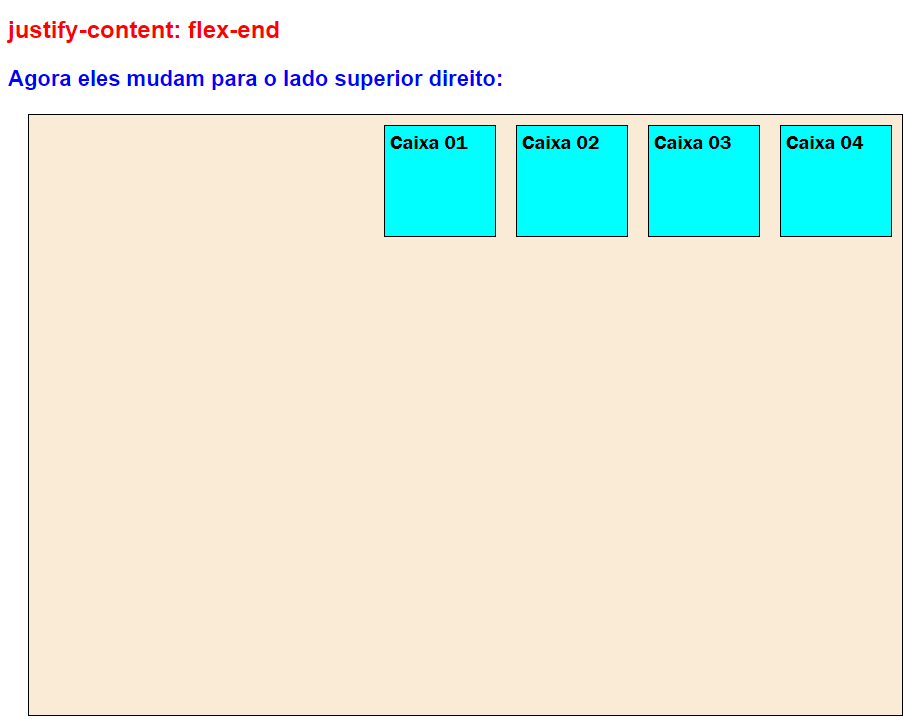
****

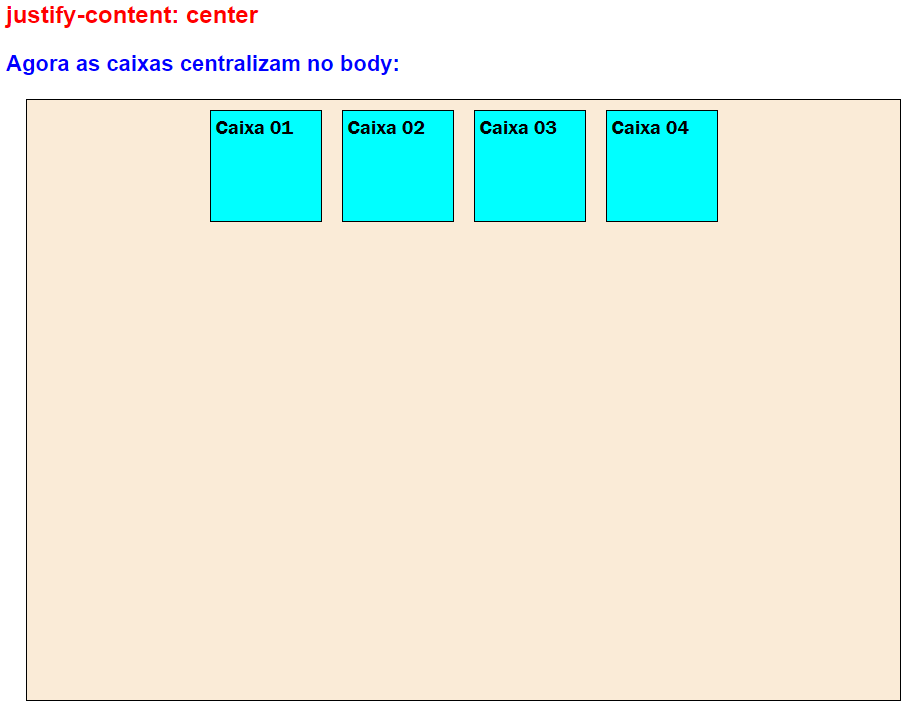
**Justify-content**

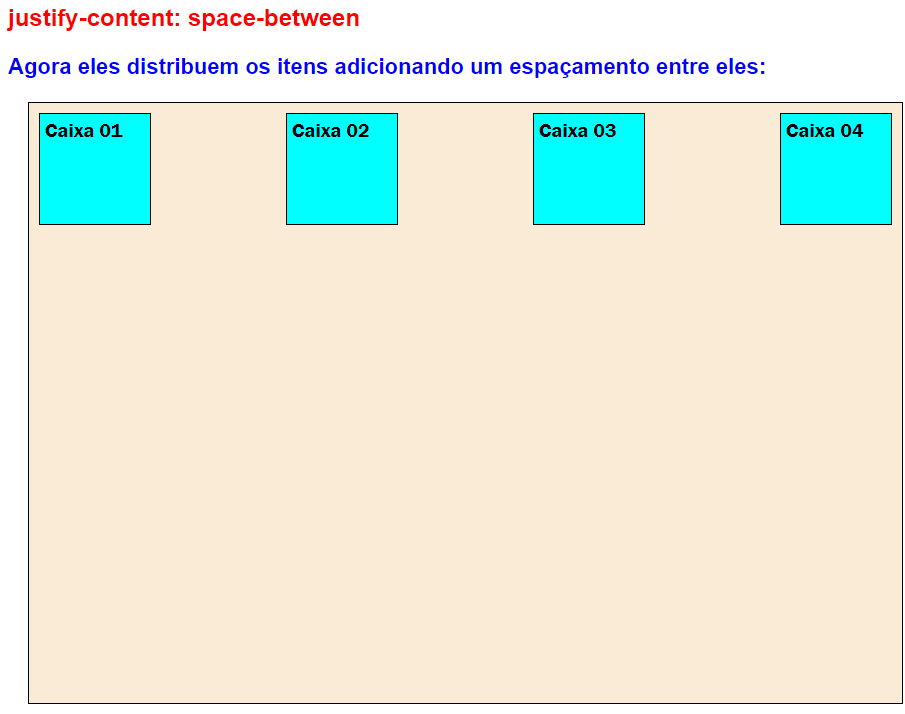
**Os valores dessa propriedade são:**

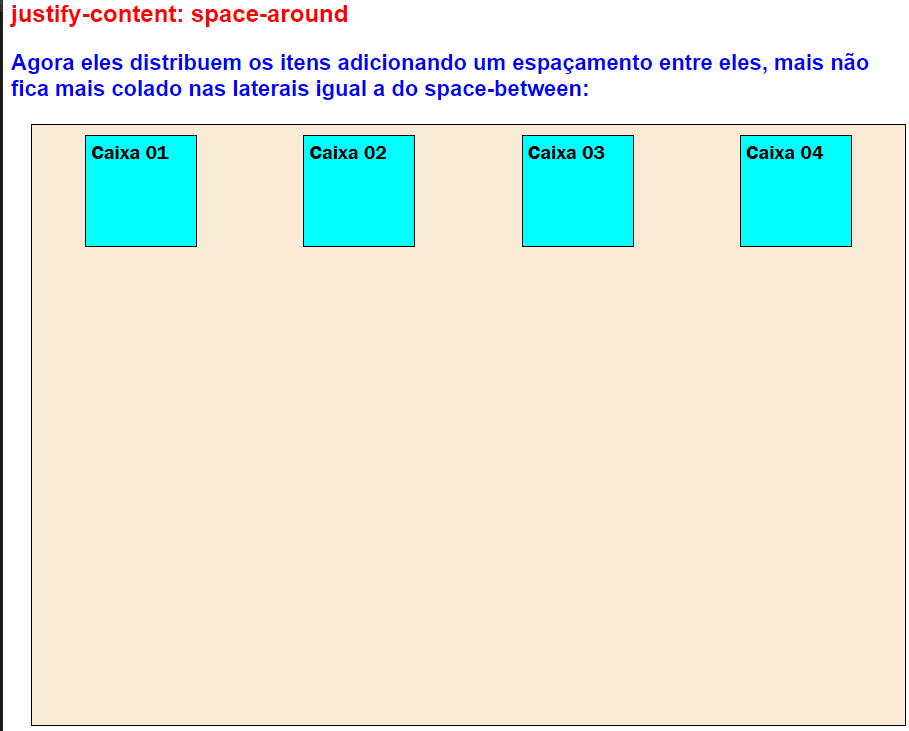
/\*flex-start| flex-end| center|space-between| space-around\*/

****

****

****

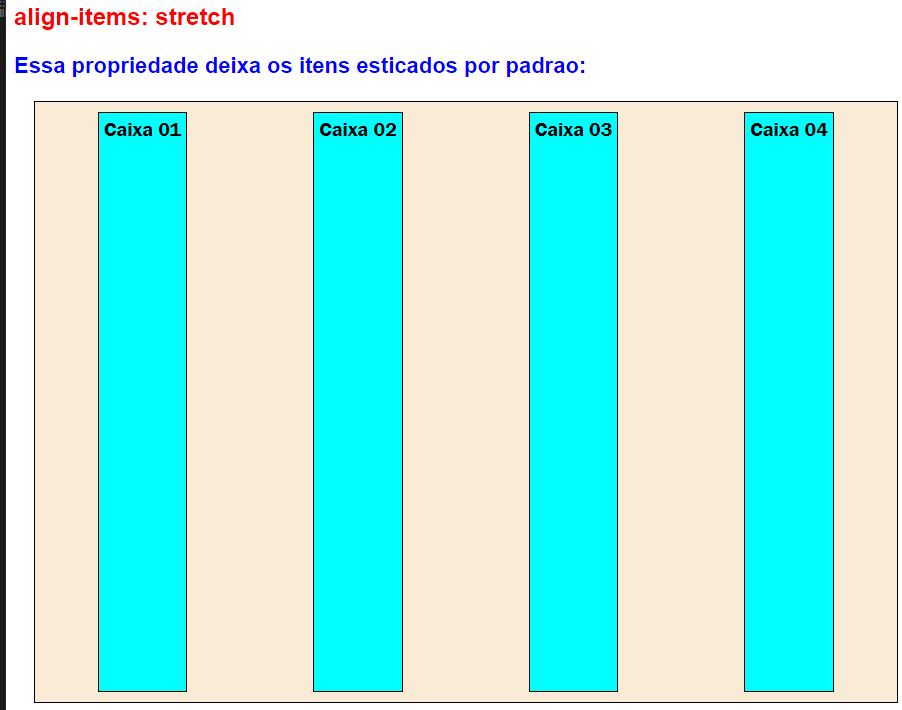
****

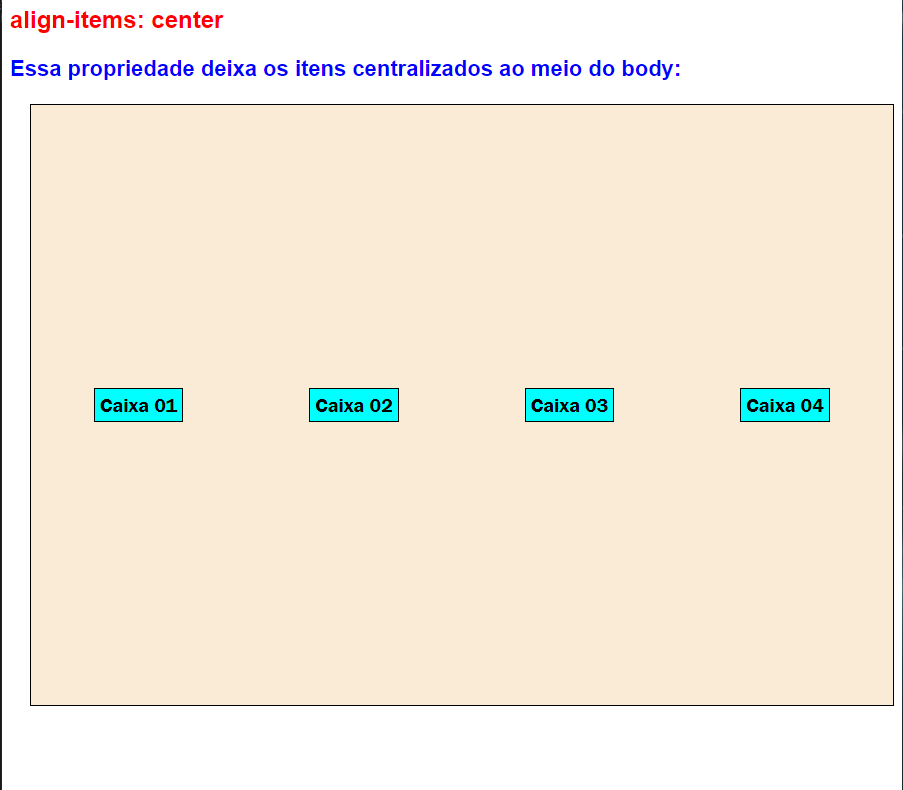
****

**Align-items:**

**Os valores dessa propriedade são:**

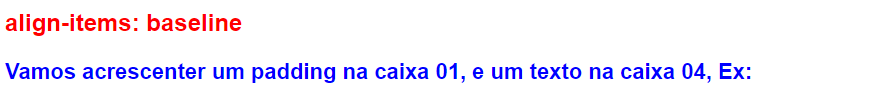
 /\*stretch| center| flex-start|flex-end| baseline\*/

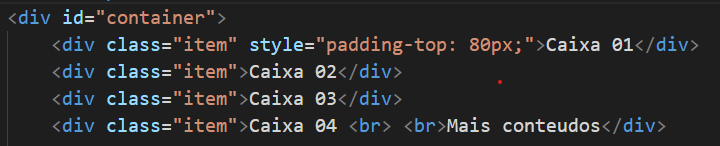
****

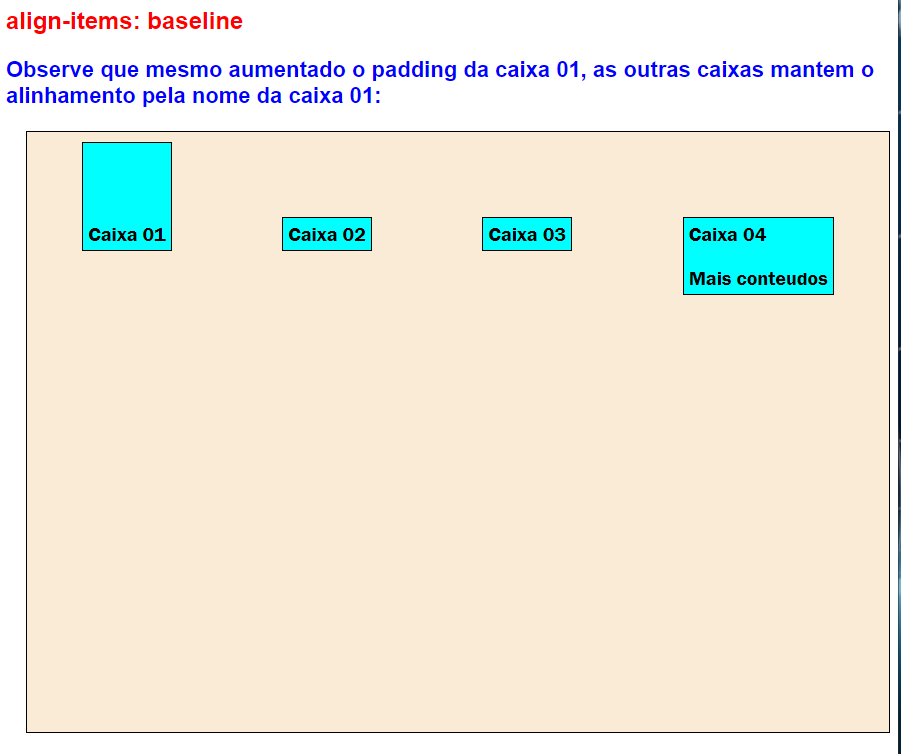
****

****

****

****

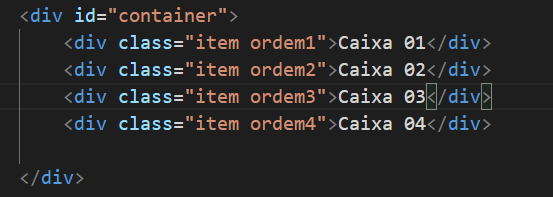
****

****

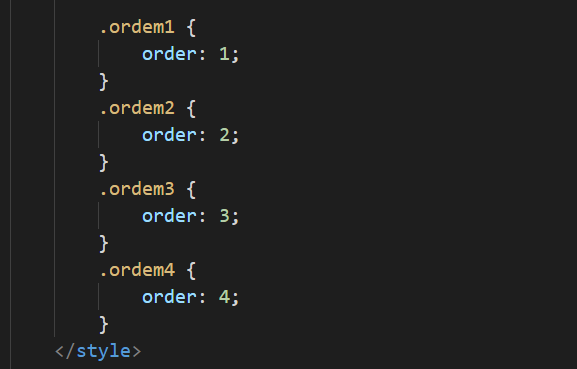
**order:**

**Mudando as ordens das caixas:**

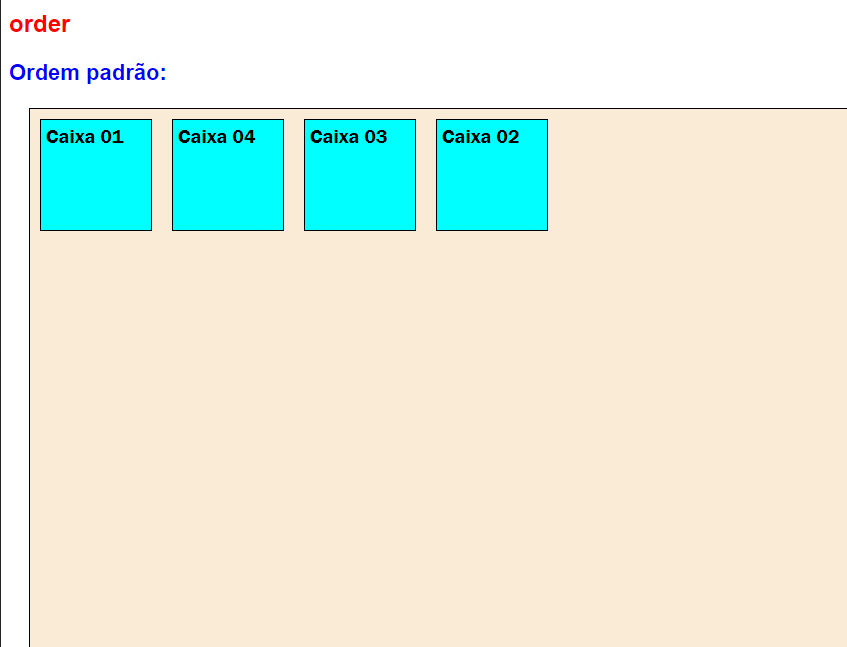
**No body vamos acrescentar na frente da class= “ordem” nas divs, ex:**

****

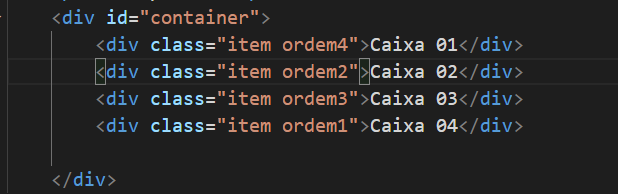
**No style:**

****

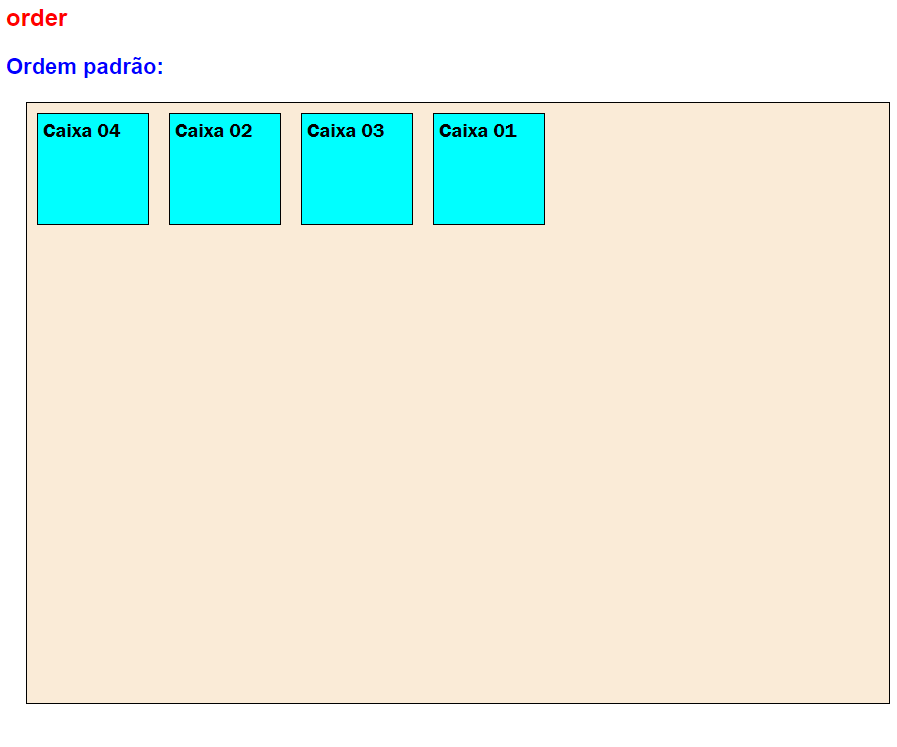
**Ficara assim, nada muda:**

****

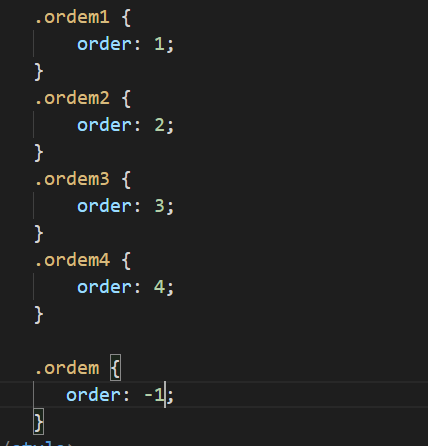
**Vamos agora inverter a ordem das classes ordem01 e ordem04, ex:**



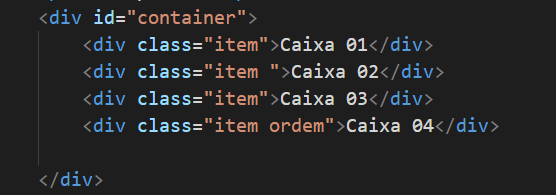
**Resultado, agora as caixas se invertem:**

****

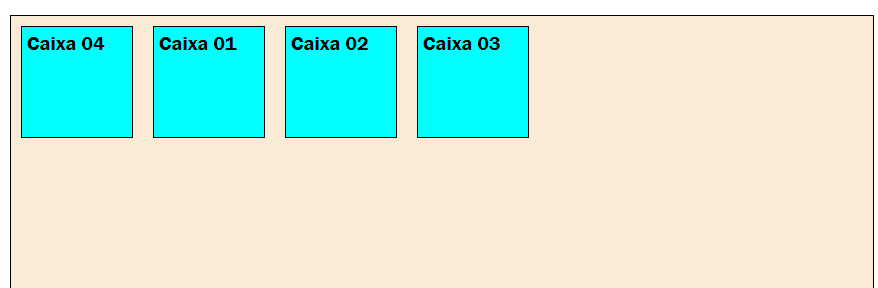
**Outro exemplo, todas as ordens tem o valor padrão 0, vamos abrir uma classe com o valor negativo, ex:**

****

**Agora vamos acrescentar essa class na caixa 04, Ex:**

****

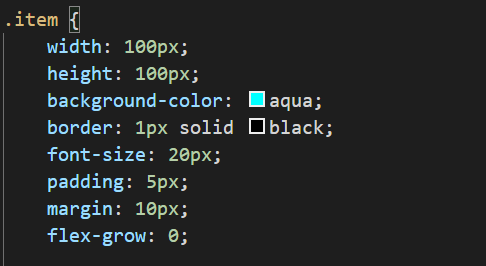
**A caixa 04 passa a tomar a primeira posição por ter adicionado uma classe negativa:**

****

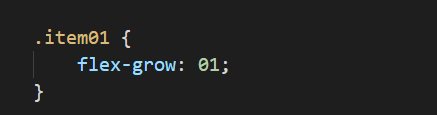
**Flex-grow:**

**Define a proporção de espaço ocupado por um item.**

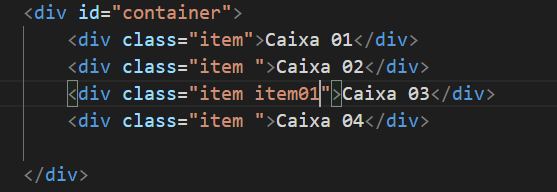
**Vamos adicionar ao seletor item o flex-grow:**

       }

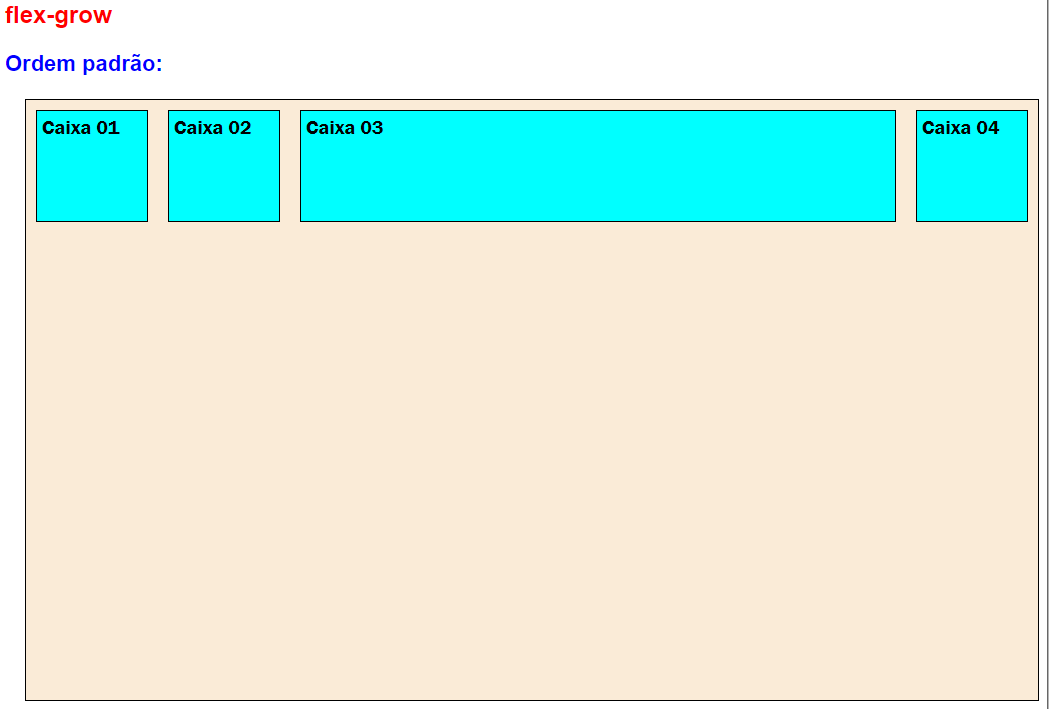
**Vamos criar uma classe agora:**

****

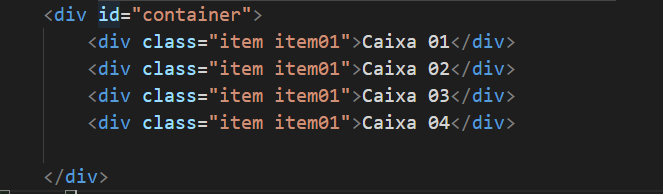
**Vamos adicionar a classe item01 na caixa03**

****

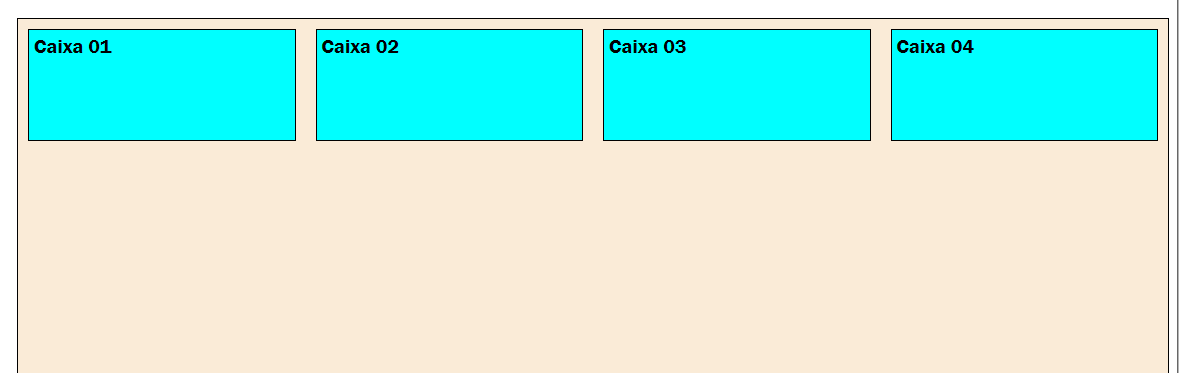
**Observe que a caixa 03 tem um tamanho diferenciado agora:**

****

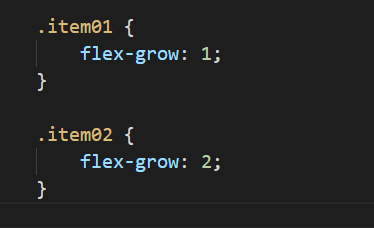
**Se adicionarmos a classe item01 em todas as caixas:**

****

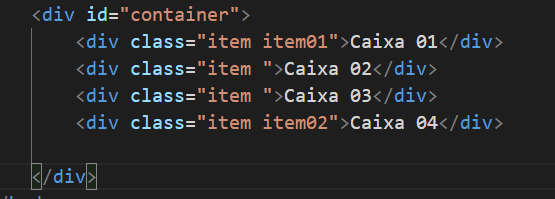
**Todos as caixas vão querer ocupar o mesmo espaço:**

****

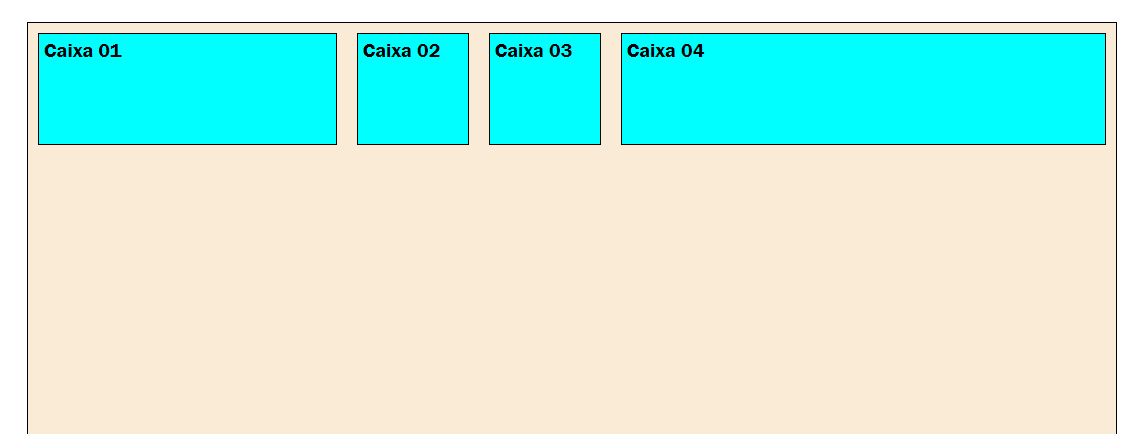
**Outro exemplo, vamos adicionar mais uma classe com o valor 2 agora:**

****

**Agora vamos adiciona-la na caixa 04:**

****

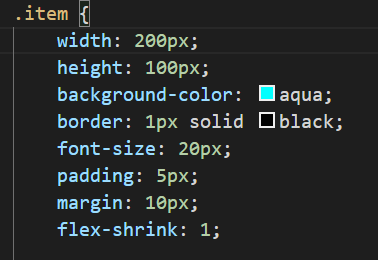
**Por a caixa04 ter o valor 2 no flex-grow que é maior do que 1 adicionada para a caixa01 ela ocupara maior espaço dentro do container:**

****

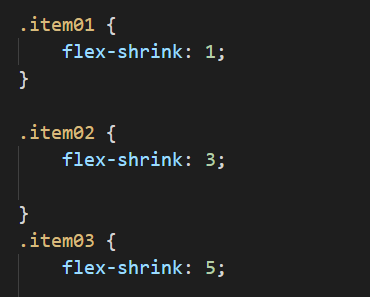
**Flex-shrink:**

**Define a capacidade de redução de tamanho do item.**

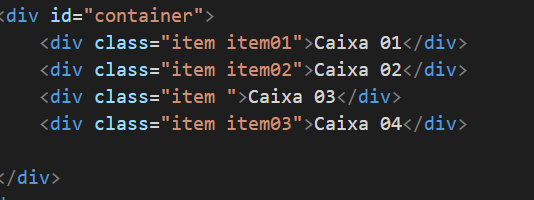
**Vamos adicionar o flex-shrink na classe item:**

****

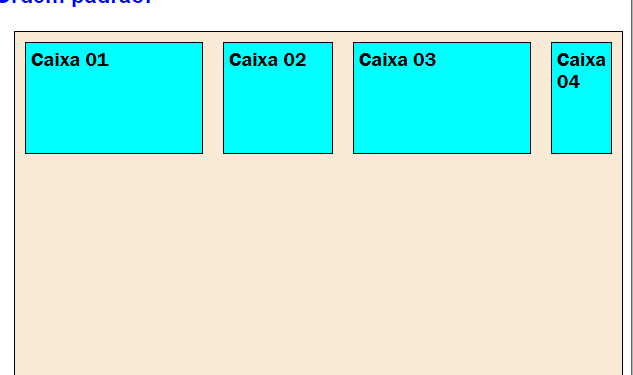
**Vamos abrir 3 classes flex-shrink com 3 valores diferentes:**

****

**Vamos adicionar essas classes nas caixas:**



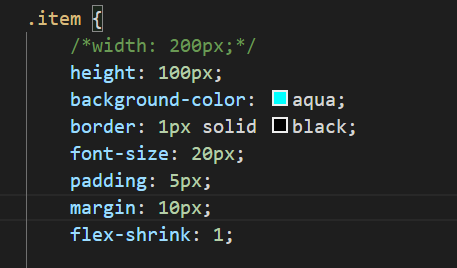
**Agora observe quando reduzimos a tela as caixas que receberam maior valor na sua flex-shrink estão sendo reduzidas, que no caso é a caixa 02 e 04:**

****

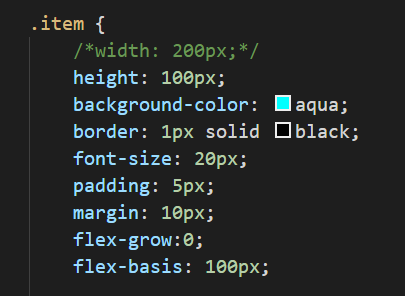
**Flex-basis:**

**Indica o tamanho inicial do flex item antes da distribuição do espaço restante.**

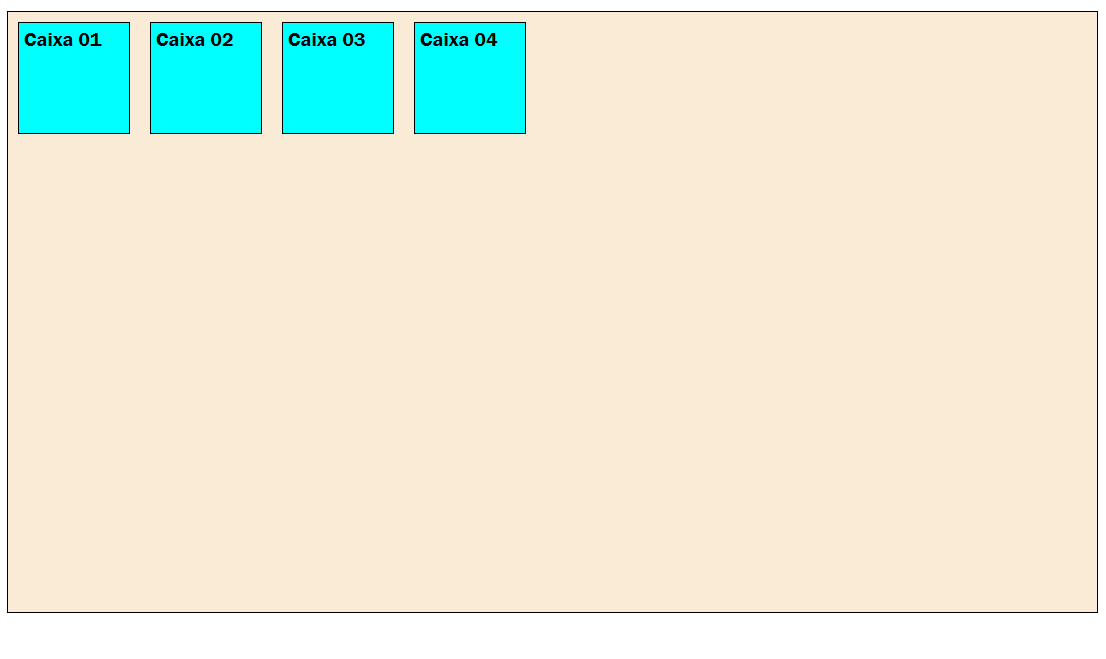
**Exemplo, vamos comentar nossa width:**

****

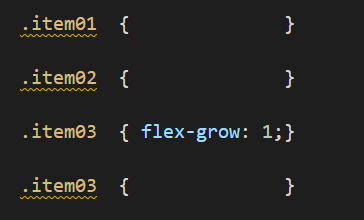
**Vamos adicionar o flex-basis no item:**

****

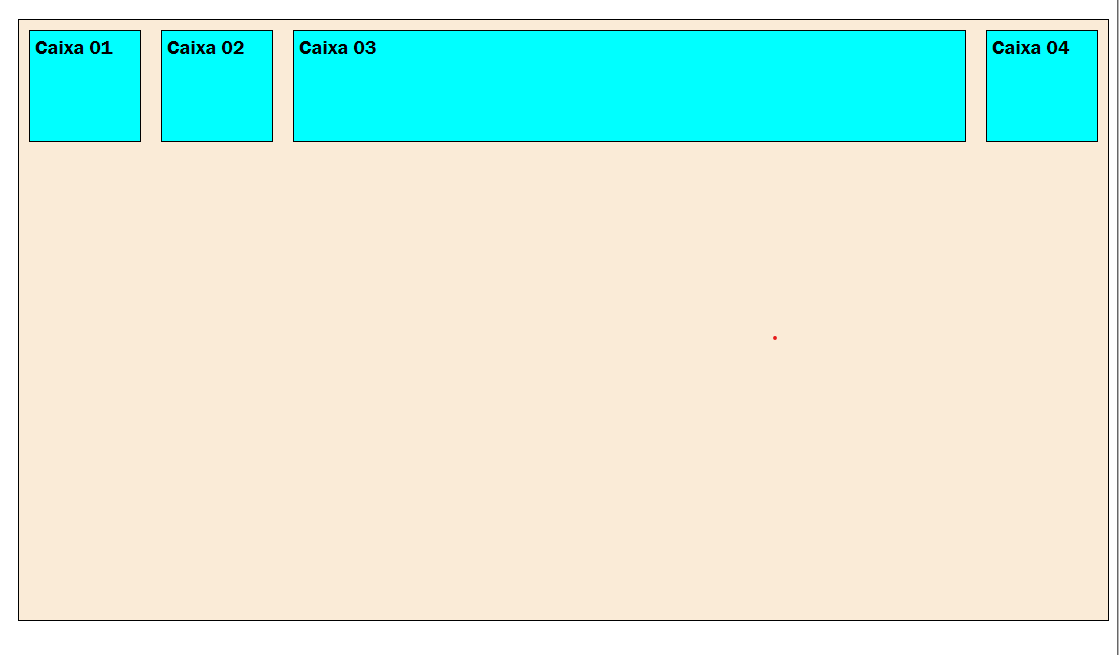
**Todas as caixas possuem 100px**

****

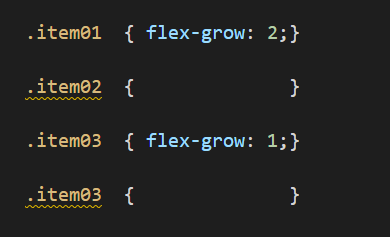
**Vamos adicionar agora um flex-grow na classe item03:**

****

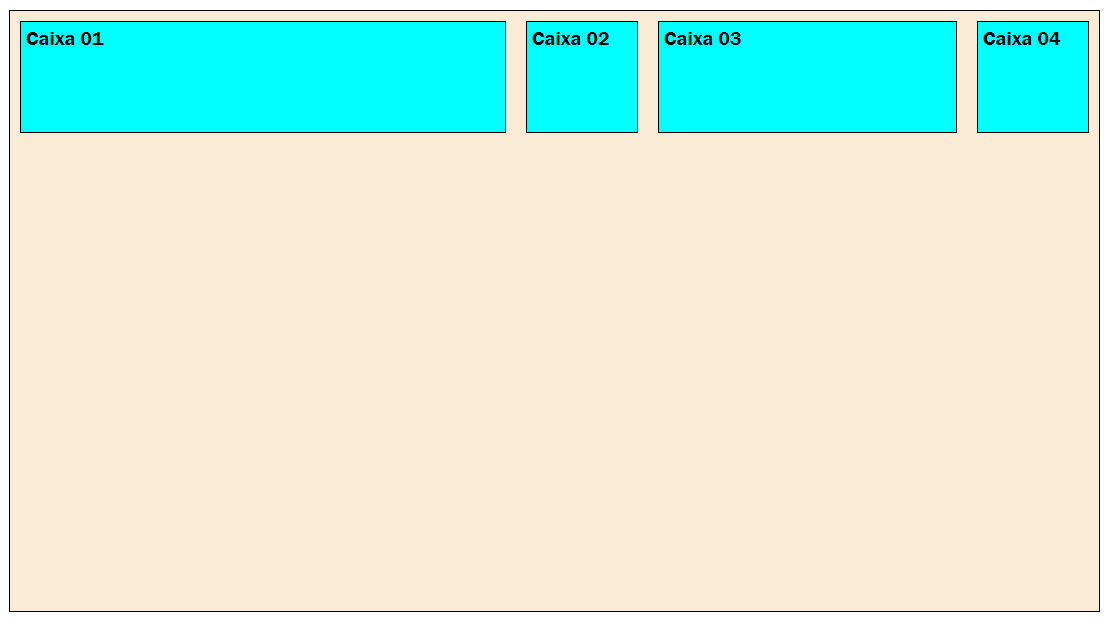
**Esse será o resultado:**

****

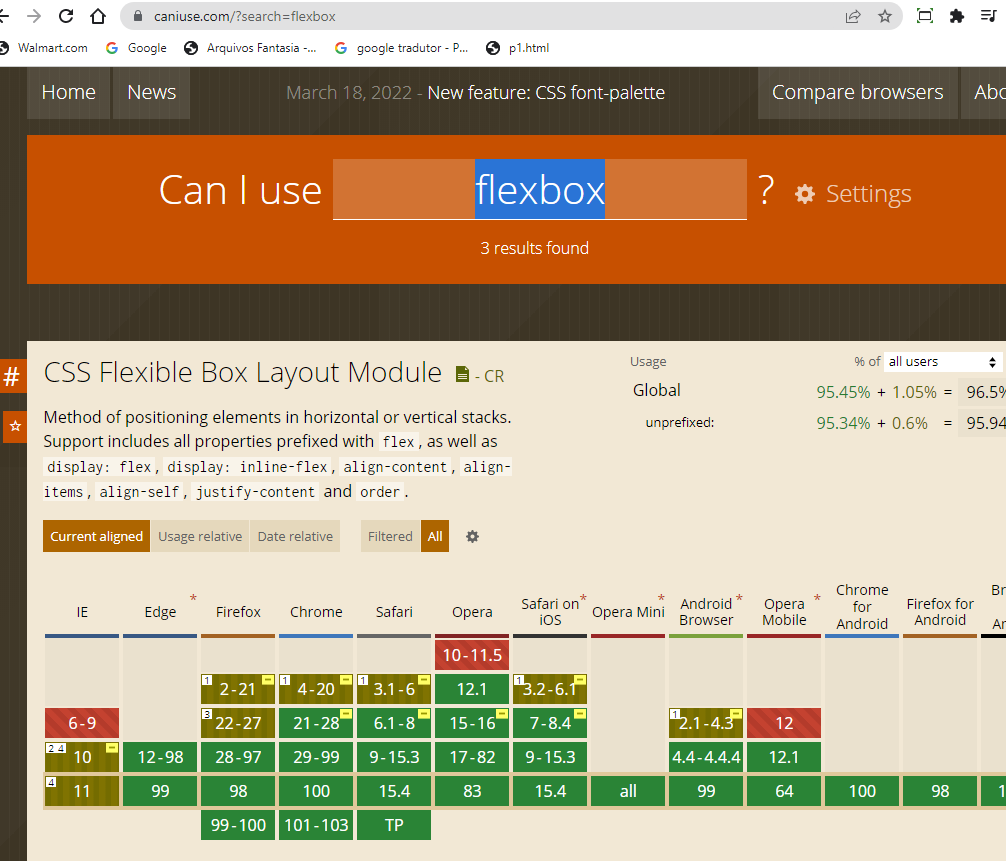
**Vamos adicionar agora um flex-grow na classe item01 e item03:**

****

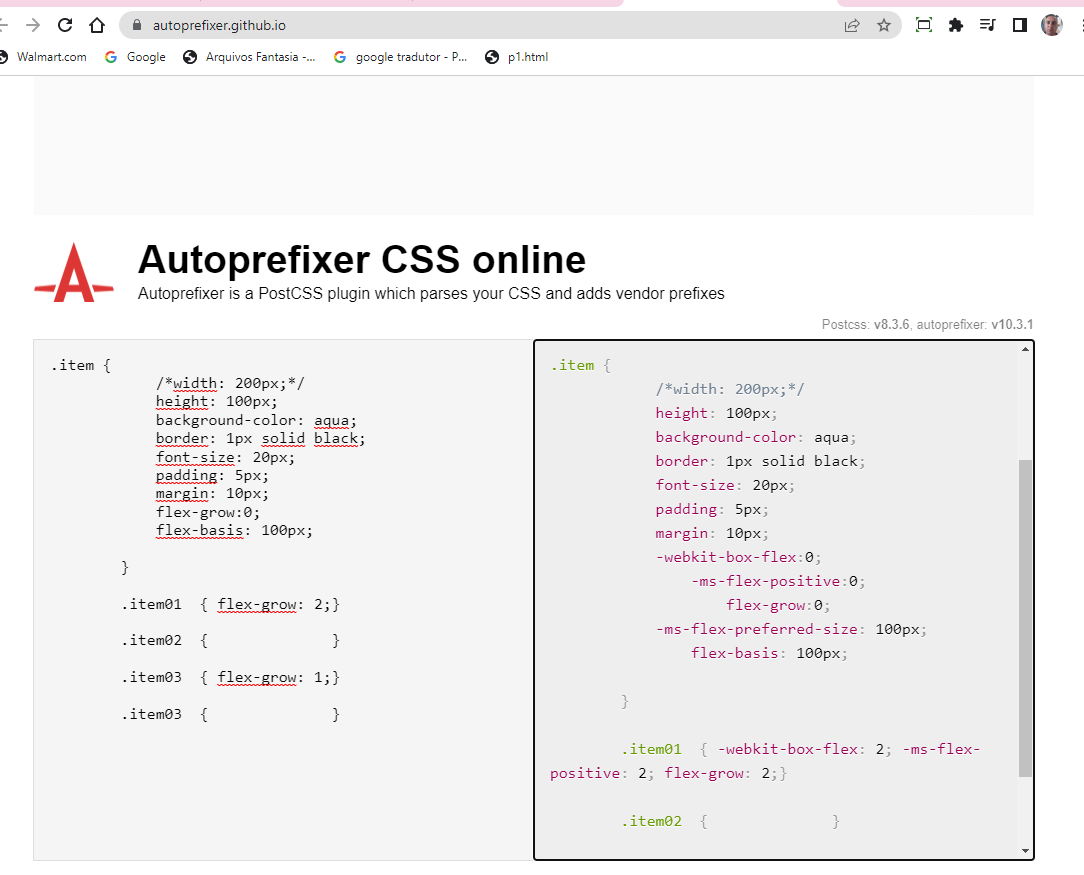
**Agora a caixa 01 e 02 começam a dividir o espaço de acordo com a proporção do flex-grow:**

****

**Esse site vera qual navegadores tem compatibilidade com o que busca:**

****

**Caso não tenha compatibilidade, entre no site autoprefixo e cole o seu código a direita, onde sera representado já com os prefixos adicionados para a compatibilidade:**

****